

**AMENDMENTS TO THE CLAIMS**

1. (Currently Amended) A method for controlling model execution in a graphical modeling environment, said method comprising:

displaying a view of an executable graphical model with a plurality of executable time-based components, said executable graphical model including at least one user-configurable, executable graphical post component having at least one input port for receiving at least one input signal, said executable graphical post component being configured to post an event when a condition associated with said at least one input signal of said executable graphical post component is satisfied;

logically associating at least one executable time-based component with said event;

identifying when said condition is satisfied during execution of said executable graphical model;

posting, using said executable graphical post component, said event by informing an event handler ~~said notice~~ of an occurrence of said event in said graphical modeling environment; ~~to an event handler;~~

~~said posting~~ notifying said at least one executable time-based component that is logically associated with said event handler of said occurrence of said event, said occurrence of said event triggering an execution of said at least one executable time-based component; and

executing said at least one executable time-based component ~~from said plurality of executable time-based components~~ in response to said notifying as opposed to in response to a specific point in time.

2. (Currently Amended) The method of claim 1, further comprising:

registering said at least one ~~of said plurality of~~ executable time-based ~~component~~components with said event handler; ~~and~~

~~receiving at said at least one of said plurality of executable time-based components registering with said event handler, notification of said occurrence of said event following said posting.~~

3. (Previously Presented) The method of claim 1 wherein said executable graphical post component is a block or label.

4. (Previously Presented) The method of claim 1, further comprising:  
    setting a sample time for an initial execution of at least one executable time-based component to be said occurrence of said specified event.
5. (Previously Presented) The method of claim 4, further comprising:  
    propagating said sample time to at least one other executable time-based component in said graphical model, said at least one other executable time-based component configured to inherit a sample rate.
6. (Previously Presented) The method of claim 4, further comprising:  
    setting a sample time of a plurality of non-contiguous executable time-based components in said graphical model to be said occurrence of said event.
7. (Previously Presented) The method of claim 6 wherein said sample time for said plurality of non-contiguous executable time-based components is set without adjusting visible connections between said executable time-based components displayed in said view.
8. (Previously Presented) The method of claim 4, further comprising:  
    indicating with an event ID in said view that said sample time of said at least one executable time-based component is set to said event.
9. (Original) The method of claim 4 wherein said event is an explicit event set by a user.
10. (Previously Presented) The method of claim 4 wherein said event is an implicit event caused by said execution of said graphical model.
11. (Previously Presented) The method of claim 10 wherein said implicit event is one of power-up, power-down and initialization.
12. (Previously Presented) The method of claim 10 wherein said implicit event corresponds to one of enabling and disabling of a subsystem.

13. (Previously Presented) The method of claim 2, further comprising:  
indicating which event an executable time-based component receives with a user-configurable color in said view.
14. (Previously Presented) The method of claim 1, wherein an execution scope of said specified event for which said execution of said graphical model is being monitored is restricted to a portion of said graphical model.
15. (Previously Presented) The method of claim 1 wherein each event in said graphical model maps on a one-to-one basis to an event handler, said event handler being a function.
16. (Original) The method of claim 15 wherein said function is inlined.
17. (Previously Presented) The method of claim 1 wherein a branch priority executable time-based component indicates an order of execution among at least two branches of said executable time-based components in response to said notifying.
18. (Previously Presented) The method of claim 1 wherein more than one executable time-based component group executes in response to said notifying, said executable time-based component groups being a user selected grouping of executable time-based components, said order of execution of said executable time-based component groups specified by a user.
19. (Currently Amended) A method for controlling model execution in a modeling environment, said method comprising:  
displaying a view of an executable model with a plurality of executable time-based components, said model including at least one user-configurable, executable graphical post component having at least one input port for receiving at least one input signal, said graphical post component being configured to post [[an]]a specified event when a condition associated with said at least one input signal of said executable graphical post component is satisfied;  
identifying when said condition is satisfied during said execution of said executable model;

posting, using said executable graphical post component, said specified event by informing an event handler ~~said notice of an occurrence of said specified event in said modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event;~~

interrupting execution of an executing event in response to said posting ~~determination of said occurrence of said specified event;~~ and

performing an operation in said executable model in response to said posting ~~determination of said occurrence of said specified event.~~

20. (Previously Presented) The method of claim 19 wherein said specified event is treated as a normal event and further comprising:

resuming execution of said interrupted event.

21. (Previously Presented) The method of claim 19 wherein said specified event is treated as an exception event and further comprising:

returning control of said execution of said model to a calling process which called said interrupted executing event without resuming execution of said interrupted event.

22. (Original) The method of claim 19 wherein said specified event is specified using an instantiated event object.

23. (Original) The method of claim 22 wherein said event is an explicit event.

24. (Original) The method of claim 22 wherein said event is an implicit event.

25. (Original) The method of claim 22 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

26. (Original) The method of claim 25 wherein said task object has at least one of a specified execution rate and priority.

27. (Previously Presented) The method of claim 26 wherein at least two events with different tasks are executing in a model and further comprising:

using event transition components to schedule said execution of time-based components associated with said at least two events, said event transition components separating said execution of said time-based components associated with said at least two events.

28. (Previously Presented) The method of claim 19 wherein said operation is controlled by an order of execution indicated in a branch priority block.

29. (Previously Presented) The method of claim 19 wherein said operation is said execution of more than one executable time-based component groups, said executable time-based component groups being a user selected grouping of executable time-based components, said order of execution of said executable time-based component groups specified by a user.

30-32. (Canceled)

33. (Currently Amended) A physical computer-readable medium holding computer-executable instructions for controlling model execution in a graphical modeling environment, said instructions comprising:

one or more instructions for displaying a view of an executable graphical model with a plurality of executable time-based components, said executable graphical model including at least one user-configurable, executable graphical post component having at least one input port for receiving at least one input signal, said executable graphical post component being configured to post an event when a condition associated with said at least one input signal of said executable graphical post component is satisfied;

one or more instructions for logically associating at least one executable time-based component with said event;

one or more instructions for identifying when said condition is satisfied during said execution of said executable graphical model;

one or more instructions for posting, using said executable graphical post component, said event by informing an event handler ~~said notice~~ of an occurrence of said event in said modeling environment; ~~to an event handler;~~

one or more instructions for ~~said posting~~ notifying said at least one executable time-based component that is logically associated with said event ~~handler~~ of said occurrence of said event, said occurrence of said event triggering an execution of said at least one executable time-based component; and

one or more instructions for executing said at least one executable time-based component ~~from said plurality of executable time-based components~~ in response to said notifying as opposed to in response to a specific point in time.

34. (Currently Amended) The medium of claim 33, wherein said instructions further comprise:

one or more instructions for registering said at least one ~~of said plurality of executable time-based components~~ component with said event handler; ~~and~~

~~one or more instructions for receiving at said at least one of said plurality of executable time-based components registering with said event handler notification of said occurrence of said event following said posting.~~

35. (Previously Presented) The medium of claim 33, wherein said executable graphical post component is a block or label.

36. (Previously Presented) The medium of claim 33, wherein said instructions further comprise:

one or more instructions for setting a sample time for an initial execution of at least one executable time-based component to be said occurrence of said specified event.

37. (Previously Presented) The medium of claim 36, wherein said instructions further comprise:

one or more instructions for propagating said sample time to at least one other executable time-based component in said graphical model, said at least one other executable time-based component configured to inherit a sample rate.

38. (Previously Presented) The medium of claim 36, wherein said instructions further comprise:

one or more instructions for setting a sample time of a plurality of non-contiguous executable time-based components in said graphical model to be said occurrence of said event.

39. (Previously Presented) The medium of claim 38 wherein said sample time for said plurality of non-contiguous executable time-based components is set without adjusting visible connections between executable time-based components displayed in said view.

40. (Previously Presented) The medium of claim 36, wherein said instructions further comprise:

one or more instructions for indicating with an event ID in said view that said sample time of said at least one executable time-based component is set to said event.

41. (Original) The medium of claim 36 wherein said event is an explicit event set by a user.

42. (Previously Presented) The medium of claim 36 wherein said event is an implicit event caused by said execution of said graphical model.

43. (Previously Presented) The medium of claim 42 wherein said implicit event is one of power-up, power-down and initialization.

44. (Previously Presented) The medium of claim 42 wherein said implicit event corresponds to one of said enabling and disabling of a subsystem.

45. (Previously Presented) The medium of claim 34, wherein said instructions further comprise:

one or more instructions for indicating which event a executable time-based component receives with a user-configurable color in said view.

46. (Previously Presented) The medium of claim 33, wherein an execution scope of said specified event for which said execution of said graphical model is being monitored is restricted to a portion of said graphical model.

47. (Previously Presented) The medium of claim 33 wherein each event in said graphical model maps on a one-to-one basis to an event handler, said event handler being a function.

48. (Original) The medium of claim 47 wherein said function is inlined.

49. (Previously Presented) The medium of claim 33 wherein a branch priority block indicates an order of execution among at least two branches of executable time-based components in response to said notifying.

50. (Previously Presented) The medium of claim 33 wherein more than one executable time-based component groups execute in response to said notifying, said executable time-based component groups being a user selected grouping of the executable time-based components, said order of execution of said executable time-based component groups specified by a user.

51. (Currently Amended) A physical computer-readable medium holding computer-executable instructions for controlling model execution, said instructions comprising:

one or more instructions for displaying a view of an executable model with a plurality of executable time-based components, said model including at least one user-configurable, executable graphical post component having at least one input port for receiving at least one input signal, said post component being configured to post [[an]]a specified event when a condition associated with said at least one input signal of said executable graphical post component is satisfied;

one or more instructions for identifying said satisfaction of said specified condition during said execution of said executable model;

one or more instructions for posting, using said executable graphical post component, said specified event by informing an event handler ~~said notice of an occurrence of said specified event in said modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event;~~

one or more instructions for interrupting execution of an executing event in response to said posting ~~determination of said occurrence of said specified event;~~ and

one or more instructions for performing an operation in said executable model in response to said posting ~~determination of said occurrence of said specified event.~~



52. (Previously Presented) The medium of claim 51 wherein said specified event is treated as a normal event and wherein said instructions further comprise:

one or more instructions for resuming execution of said interrupted event.

53. (Previously Presented) The medium of claim 51 wherein said specified event is treated as an exception event and wherein said instructions further comprise:

one or more instructions for returning control of said execution of said model to a calling process which called said interrupted executing event without resuming execution of said interrupted event.

54. (Original) The medium of claim 51 wherein said specified event is specified using an instantiated event object.

55. (Original) The medium of claim 54 wherein said event is an explicit event.

56. (Original) The medium of claim 54 wherein said event is an implicit event.

57. (Original) The medium of claim 54 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

58. (Original) The medium of claim 57 wherein said task object has at least one of a specified execution rate and priority.

59. (Previously Presented) The medium of claim 58 wherein at least two events with different tasks are executing in a model and wherein said instructions further comprise:

one or more instructions for using event transition components to schedule said execution of executable time-based components associated with said at least two events, said event transition components separating said execution of said executable time-based components associated with said at least two events.

60. (Previously Presented) The medium of claim 51 wherein said operation is controlled by an order of execution indicated a branch priority block.

61. (Previously Presented) The medium of claim 51 wherein said operation is said execution of more than one executable time-based component groups, said executable time-based component groups being a user selected grouping of said executable time-based components, said order of execution of said executable time-based component groups specified by a user.